



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/456,894	12/07/1999	CLINTON EDWARD LUM	003845.P002	7047

7590 12/16/2005

W. SCOTT PETTY
KING & SPALDING
191 PEACHTREE STREET
45TH FLOOR
ATLANTA, GA 30303-1763

EXAMINER

EL CHANTI, HUSSEIN A

ART UNIT PAPER NUMBER

2157

DATE MAILED: 12/16/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/456,894	Applicant(s) LUM, CLINTON EDWARD	
	Examiner Hussein A. El-chanti	Art Unit 2157	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 20 September 2005.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-26,28 and 29 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-26,28 and 29 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date: _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date: _____ | 6) <input type="checkbox"/> Other: _____ |

Response to Amendment

1. This action is responsive to amendment received on Sep. 20, 2005. Claims 1, 4, 5, 9-12, 14, 16-18, 22-26 and 28 were amended. Claims 1, 2, 4-12, 14-26 and 28-29 are pending examination.

Information Disclosure Statement

2. The information disclosure statement filed Oct. 11, 2005 fails to comply with 37 CFR 1.98(a)(3) because it does not include a concise explanation of the relevance, as it is presently understood by the individual designated in 37 CFR 1.56(c) most knowledgeable about the content of the information, of each patent listed that is not in the English language. It has been placed in the application file, but the information referred to therein has not been considered.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 1, 2, 4-12, 14-26 and 28-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fletcher et al., U.S. Patent No. 6,009,274 (referred to hereafter as Fletcher) in view of Johnston et al., U.S. Patent No. 5,781,776 (referred to hereafter as Johnston).

As to claim 1, Fletcher teaches a machine readable-medium having stored thereon sequences of instructions which when executed by a processor cause the processor to perform the acts of:

disabling access to at least a first section of computer code in a network driver software interface that is being executed by the processor by overwriting computer code that is executed before the first section of computer code with blocking computer code wherein the network driver software interface provides for communication between one or more media access control units and one or more protocol drivers in a computer system according to a set of bindings (see col. 9 lines 4-42 and col. 13 lines 40-55, the NIC driver is disabled);

executing the blocking code with the processor (see col. 9 lines 4-42 and col. 13 lines 40-55);

patching the first section of computer code while the blocking computer code of the software is being executed by the processor, the patching of the first section of code causing the insertion of a rerouting driver into the one or more communication paths provided by the set of bindings (see col. 9 lines 4-42 and col. 13 lines 40-55, the driver is updated without rebooting the machine where the update is done in a transparent manner to the client); and

re-enabling access to the patched first section of computer code by replacing the blocking computer code with computer code that allows execution of the patched first section of computer code (see col. 9 lines 4-42 and col. 13 lines 40-55, the updated driver is then re-enabled).

Fletcher does not explicitly teach "inserting a template jump to another template". However Johnston teaches a system and method for replacing or adding instructions in a computer code by inserting a template jump to an updated piece of code (see col. 8 lines 39-col. 9 lines 4). It would have been obvious for one of the ordinary skill in the art at the time of the invention to modify Fletcher by inserting a template jump in the Network interface driver to update the driver as taught by Johnston because doing so would to avoid errors in the control code when instructions are moved as suggested by Johnston (see Johnston col. 8 lines 65-col. 9 lines 15).

As to claim 11, Fletcher teaches a computer implemented method comprising:

transmitting from a remote host to a first target computer on a network an installation application and a rerouting driver (see col. 9 lines 4-42 and col. 13 lines 40-55);

transmitting from the remote host to the first target computer a command to cause the first target computer to execute the installation application (see col. 9 lines 4-42 and col. 13 lines 40-55);

the first target computer, responsive to receipt of the command, executing the installation application, wherein the first target computer includes a network driver interface that provides for communication between one or more media access control units and one or more protocol drivers according to a set of bindings (see col. 9 lines 4-42 and col. 13 lines 40-55); and

the first target computer responsive to executing the installation application causing the modification of the network driver interface to insert the rerouting

driver into the one or more communication paths provided by the set of bindings without restarting the first target computer (see col. 9 lines 4-42 and col. 13 lines 40-55), the first target computer comprising a multiprocessor system wherein the insert of rerouting driver further comprises:

disabling access to at least a first section of computer code in a network driver software interface that is being executed by the processor by overwriting computer code prior to the first section with blocking code (see col. 9 lines 4-42 and col. 13 lines 40-55, the NIC driver is disabled).

Fletcher does not explicitly teach "inserting a template jump to another template". However Johnston teaches a system and method for replacing or adding instructions in a computer code by inserting a template jump to an updated piece of code (see col. 8 lines 39-col. 9 lines 4). It would have been obvious for one of the ordinary skill in the art at the time of the invention to modify Fletcher by inserting a template jump in the Network interface driver to update the driver as taught by Johnston because doing so would to avoid errors in the control code when instructions are moved as suggested by Johnston (see Johnston col. 8 lines 65-col. 9 lines 15).

As to claim 18, Fletcher teaches a computer system comprising:

- a protocol driver (see col. 9 lines 4-42 and col. 13 lines 40-55);
- a media access control unit (see col. 9 lines 4-42 and col. 13 lines 40-55);
- a rerouting driver wherein during the installation of the rerouting driver, a first section of code in the driver interface is disabled by overwriting code that is positioned before the first section of code with blocking code and wherein the first

section of code is then patched (see col. 9 lines 4-42 and col. 13 lines 40-55, the driver is updated without rebooting the machine where the update is done in a transparent manner to the client);

a network driver interface to store a first binding defining a communication path between the protocol driver and the media access control unit, the network driver interface coupled to communicate packets with the media access control unit, the network driver interface patched to communicate the packets with a rerouting driver (see col. 9 lines 4-42 and col. 13 lines 40-55); and

the rerouting driver being coupled to communicate the packets with the protocol driver (see col. 9 lines 4-42 and col. 13 lines 40-55).

Fletcher does not explicitly teach "inserting a template jump to another template". However Johnston teaches a system and method for replacing or adding instructions in a computer code by inserting a template jump to an updated piece of code (see col. 8 lines 39-col. 9 lines 4). It would have been obvious for one of the ordinary skill in the art at the time of the invention to modify Fletcher by inserting a template jump in the Network interface driver to update the driver as taught by Johnston because doing so would to avoid errors in the control code when instructions are moved as suggested by Johnston (see Johnston col. 8 lines 65-col. 9 lines 15).

As to claim 23, Fletcher teaches a rerouting driver for remotely installing network drivers and software without restarting the computer system following installation, the computer system having an operating system in which a network driver interface provides communication of information between at least one media access control unit

and at least one protocol driver on the computer system, the rerouting driver comprising:

- control code, for controlling the rerouting driver (see col. 9 lines 4-42 and col. 13 lines 40-55).;

- binding code, for establishing at least one binding at the network driver interface so that the rerouting driver is bound to at least one media access control unit (see col. 9 lines 4-42 and col. 13 lines 40-55).;

- patching code so that information from at least one media access control unit destined for at least one protocol driver is rerouted to the rerouting driver (see col. 9 lines 4-42 and col. 13 lines 40-55, the patching of the bindings is performed and the packets are rerouted according to the new set of bindings).

Fletcher does not explicitly teach "inserting a template jump to another template". However Johnston teaches a system and method for replacing or adding instructions in a computer code by inserting a template jump to an updated piece of code (see col. 8 lines 39-col. 9 lines 4). It would have been obvious for one of the ordinary skill in the art at the time of the invention to modify Fletcher by inserting a template jump in the Network interface driver to update the driver as taught by Johnston because doing so would to avoid errors in the control code when instructions are moved as suggested by Johnston (see Johnston col. 8 lines 65-col. 9 lines 15).

As to claim 26, Fletcher teaches a method for disabling and re-enabling access to code in a multiprocessor system having a shared memory and a network driver interface comprising:

selecting a first section of code in a first central processing unit that is to be modified (see col. 9 lines 4-42 and col. 13 lines 40-55, the processor being any of the gateway interface device);

writing the first section of code into the cache memory of the first central processing unit (see col. 9 lines 4-42 and col. 13 lines 40-55);

overwriting a portion of the first section of code in cache memory with blocking code to create a first version of code (see col. 9 lines 4-42 and col. 13 lines 40-55);

writing the first version of code into shared memory (see col. 9 lines 4-42 and col. 13 lines 40-55, shared memory being any of the gateway interface device);

modifying the first version of code in the cache memory to create a second version of code, wherein a portion of the code following the blocking code is overwritten to effect a static patch of the network driver interface (see col. 9 lines 4-42 and col. 13 lines 40-55);

writing the second version of code into shared memory (see col. 9 lines 4-42 and col. 13 lines 40-55);

modifying the second version of code in the cache memory with code to create a third version of code, wherein the blocking code is overwritten to remove the blocking code (see col. 9 lines 4-42 and col. 13 lines 40-55, the patching of the bindings is performed and the packets are rerouted according to the new set of bindings); and

writing the third version of code into shared memory (see col. 9 lines 4-42 and col. 13 lines 40-55).

Fletcher does not explicitly teach "inserting a template jump to another template". However Johnston teaches a system and method for replacing or adding instructions in a computer code by inserting a template jump to an updated piece of code (see col. 8 lines 39-col. 9 lines 4). It would have been obvious for one of the ordinary skill in the art at the time of the invention to modify Fletcher by inserting a template jump in the Network interface driver to update the driver as taught by Johnston because doing so would to avoid errors in the control code when instructions are moved as suggested by Johnston (see Johnston col. 8 lines 65-col. 9 lines 15).

As to claim 28, Fletcher teaches a machine readable medium having stored therein instructions which when executed cause a set of one or more processors to perform the following:

disabling access to a first section of code, the first section of code to be executed when to provide a communication path between a media access control unit and an application the first section of code including a generic call (see col. 9 lines 4-42 and col. 13 lines 40-55, the NIC is disabled); and

overwriting the first section of code with a second section of code whose execution causes execution flow to be rerouted to a third section of code in a rerouting driver, the second section of code being larger than the first section of code, the third section of code when executed completing the communication path and returning execution flow, (see col. 9 lines 4-42 and col. 13 lines 40-55, the patching of the bindings is performed and the packets are rerouted according to the new set of bindings).

Fletcher does not explicitly teach “the third section of code including additional code not present in the first section of code that is now inserted into the communication path”. However Johnston teaches a system and method for replacing or adding instructions in a computer code by inserting a template jump to an updated piece of code (see col. 8 lines 39-col. 9 lines 4). It would have been obvious for one of the ordinary skill in the art at the time of the invention to modify Fletcher by inserting a template jump in the Network interface driver to update the driver as taught by Johnston because doing so would to avoid errors in the control code when instructions are moved as suggested by Johnston (see Johnston col. 8 lines 65-col. 9 lines 15).

As to claims 2, 12 and 19, Fletcher teaches the machine readable-medium, method and system of claims 1, 11 and 19 respectively wherein the patching is static patching (see col. 9 lines 4-42 and col. 13 lines 40-55).

As to claims 3 and 13, Johnston teaches the machine readable-medium and method of claims 2 and 12 respectively wherein the static patching includes inserting a template jump from the network driver interface to a template in the rerouting driver (see col. 8 lines 39-col. 9 lines 15).

As to claims 4 and 14, Johnston teaches the machine readable-medium and method of claims 3 and 13 respectively wherein the template jumps are inserted in the code so that a CALL instruction to the code is replaced with a JUMP to the template in the rerouting driver, the template containing the CALL instruction (see col. 8 lines 39-col. 9 lines 15).

As to claim 5, Fletcher teaches the machine readable-medium of claim 2 wherein the patching the first section of code creates at least one new binding between the network driver interface and the rerouting driver (see col. 9 lines 4-42 and col. 13 lines 40-55).

As to claim 6, Fletcher teaches the machine readable-medium of claim 5 wherein the at least one new binding provides for communication between one or more media access control units and a capturing unit in the rerouting driver (see col. 9 lines 4-42 and col. 13 lines 40-55).

As to claim 7, Fletcher teaches the machine readable-medium of claim 6 wherein the capturing unit is used to intercept communications over the at least one new binding (see col. 9 lines 4-42 and col. 13 lines 40-55).

As to claims 8, 15 and 20, Fletcher teaches the machine readable-medium, method and system of claims 1, 11 and 18 respectively wherein the patching is dynamic patching (col. 13 lines 40-55).

As to claims 9 and 16, Johnston teaches inserting a template jump in the network driver interface to a template in the patching code (see col. 8 lines 39-col. 9 lines 15).

As to claims 10 and 17, Johnston teaches the machine readable-medium and method of claims 9 and 16 respectively wherein the template jumps are inserted in the code so that a CALL instruction to the code is replaced with a JUMP to the template in the code, the template containing the CALL instruction (see col. 8 lines 39-col. 9 lines 15).

As to claim 21, Fletcher teaches the computer system of claim 18 wherein the rerouting driver further comprising a capture unit to store in a buffer one or more of the packets for evaluation (see col. 9 lines 4-42).

As to claim 22, Fletcher teaches the computer system of claim 18 wherein the network interface to also store a second binding defining a communication path between the rerouting driver and the media access control unit and the capture unit to store in the buffer the packets destined for the rerouting driver (see col. 13 lines 15-50).

As to claim 24, Fletcher teaches the rerouting driver of claim 23 wherein the control code identifies a starting memory address of the network driver interface instruction code and disables access to the first section of code and further wherein the patching code following the disabling access operates to overwrite the first section of code and additional pre-determined memory addresses so the all the pre-determined memory addresses are patched (col. 13 lines 15-50, the NIC is disabled).

As to claim 25, Fletcher teaches the rerouting driver of claim 23 wherein the patching code responsive to receipt of information being sent from the network driver interface determines the instruction code address that sent the information and overwrites the first section of code at that address so that memory addresses are incrementally patched as information is received from the network driver interface (see col. 13 lines 15-50).

As to claim 29, Johnston teaches the machine-readable medium of claim 28 wherein the second section of code contains a template jump to a template in the third section code (see col. 8 lines 40-col. 9 lines 15).

4. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.
5. Applicants arguments have been considered but are moot in view of the new grounds of rejection.
6. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the date of this final action.

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Hussein A. El-chanti whose telephone number is (571)272-3999. The examiner can normally be reached on Mon-Fri 8:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ario Etienne can be reached on (571)272-4001. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Art Unit: 2157

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Hussein El-chanti

Dec. 6, 2005



ABDULLAH SALAD
PRIMARY EXAMINER